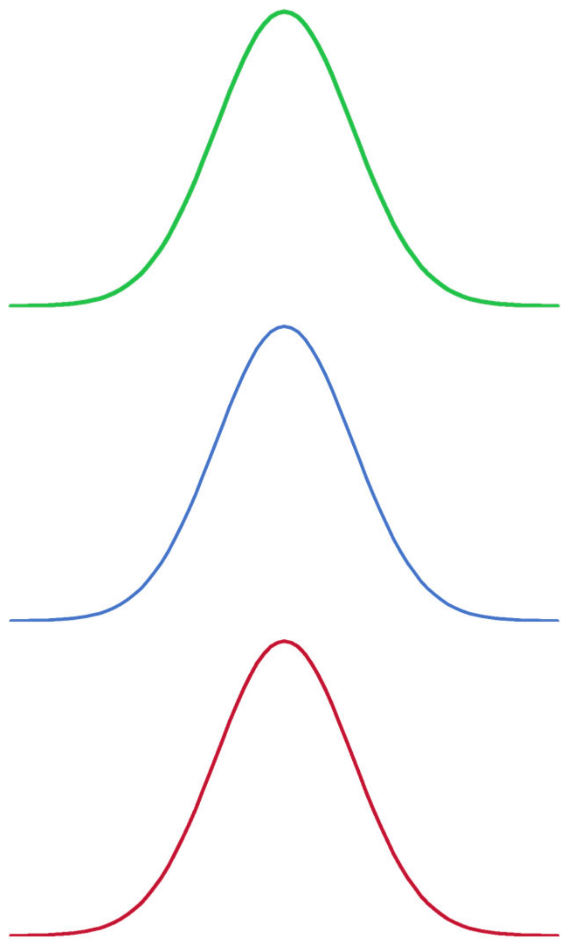
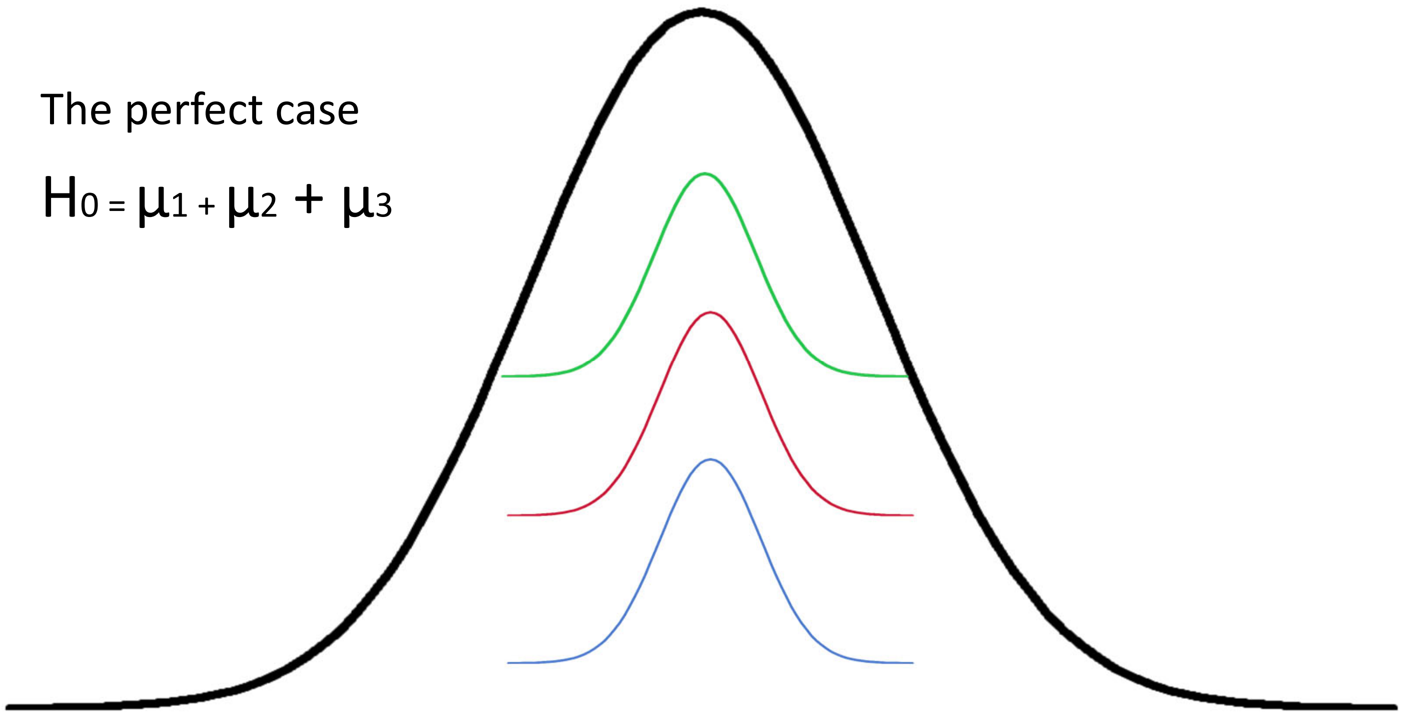
ANOVA

- An Analysis of Variance
  - Compare the means of more than two groups, samples, populations



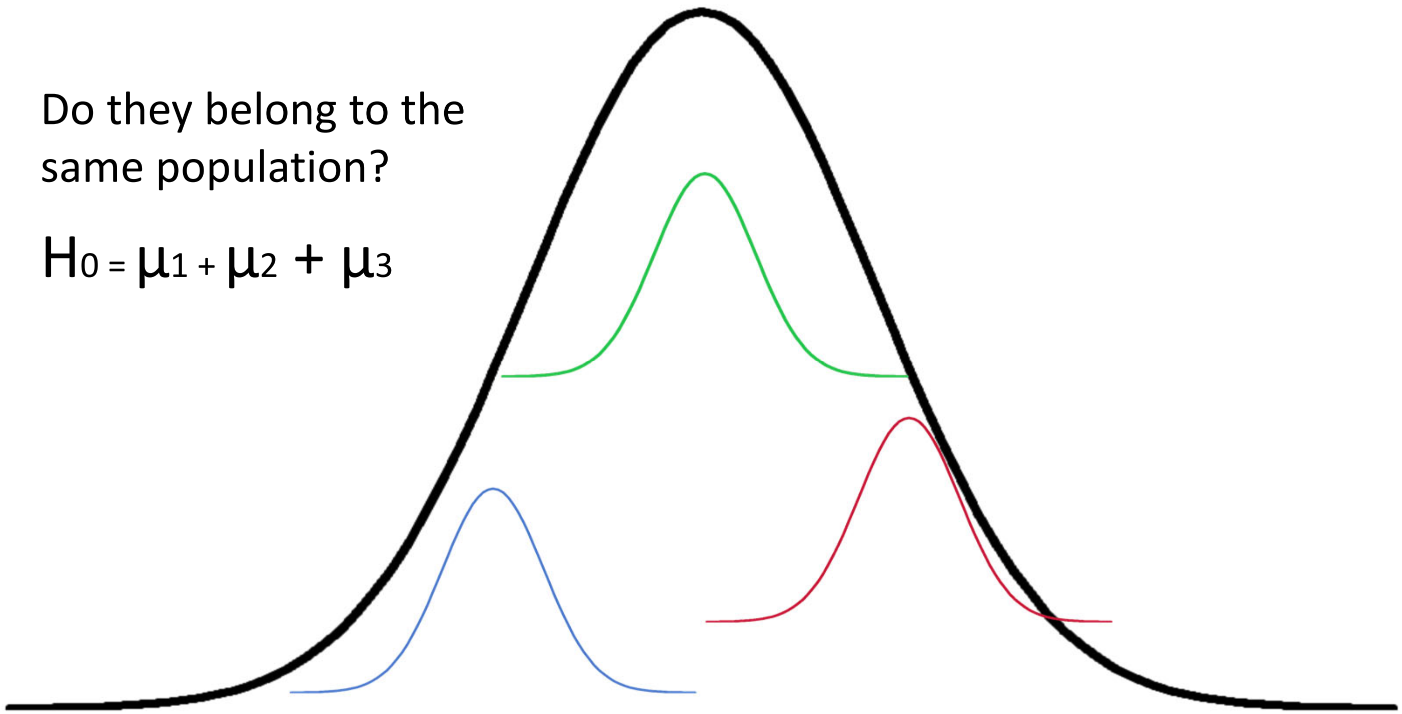
The perfect case

$H_0 = \mu_1 + \mu_2 + \mu_3$



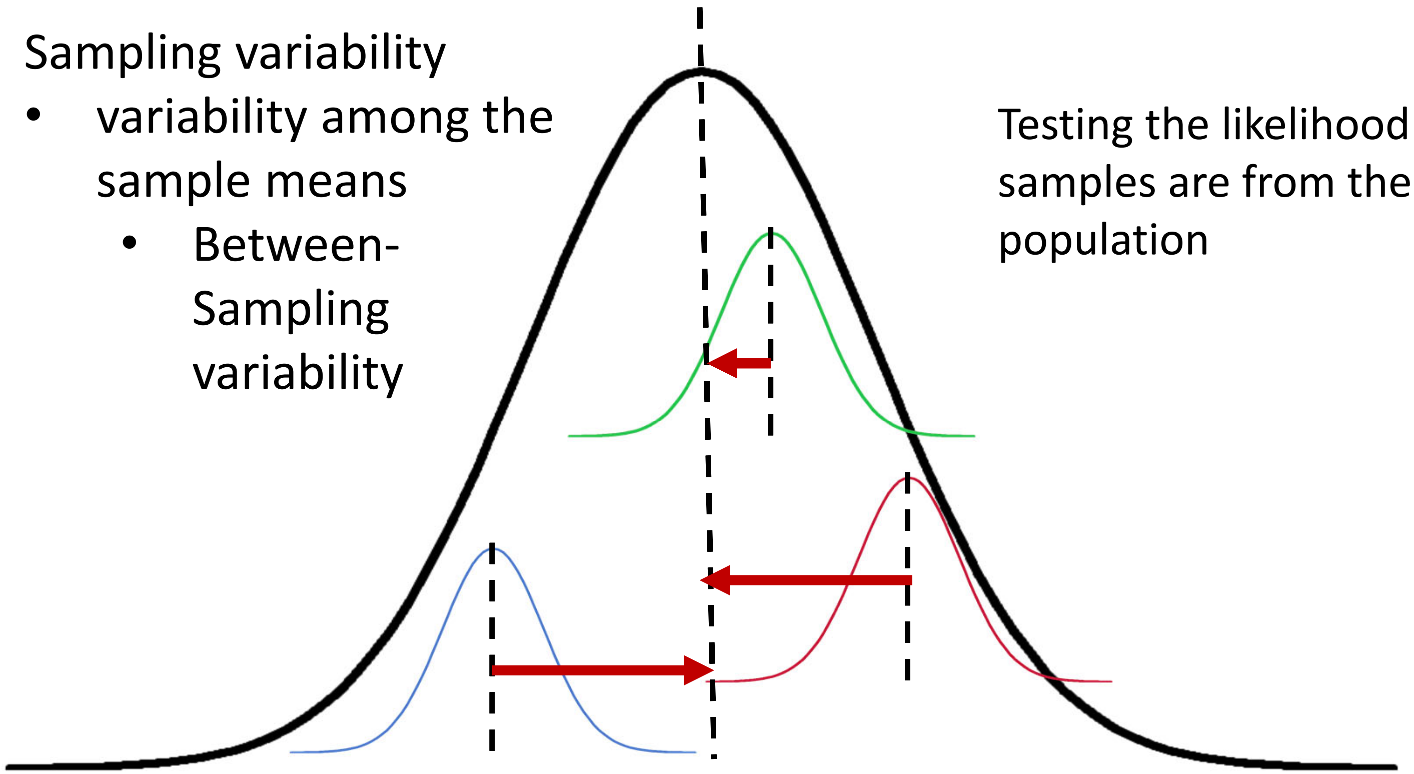
Do they belong to the same population?

$H_0 = \mu_1 + \mu_2 + \mu_3$



Sampling variability

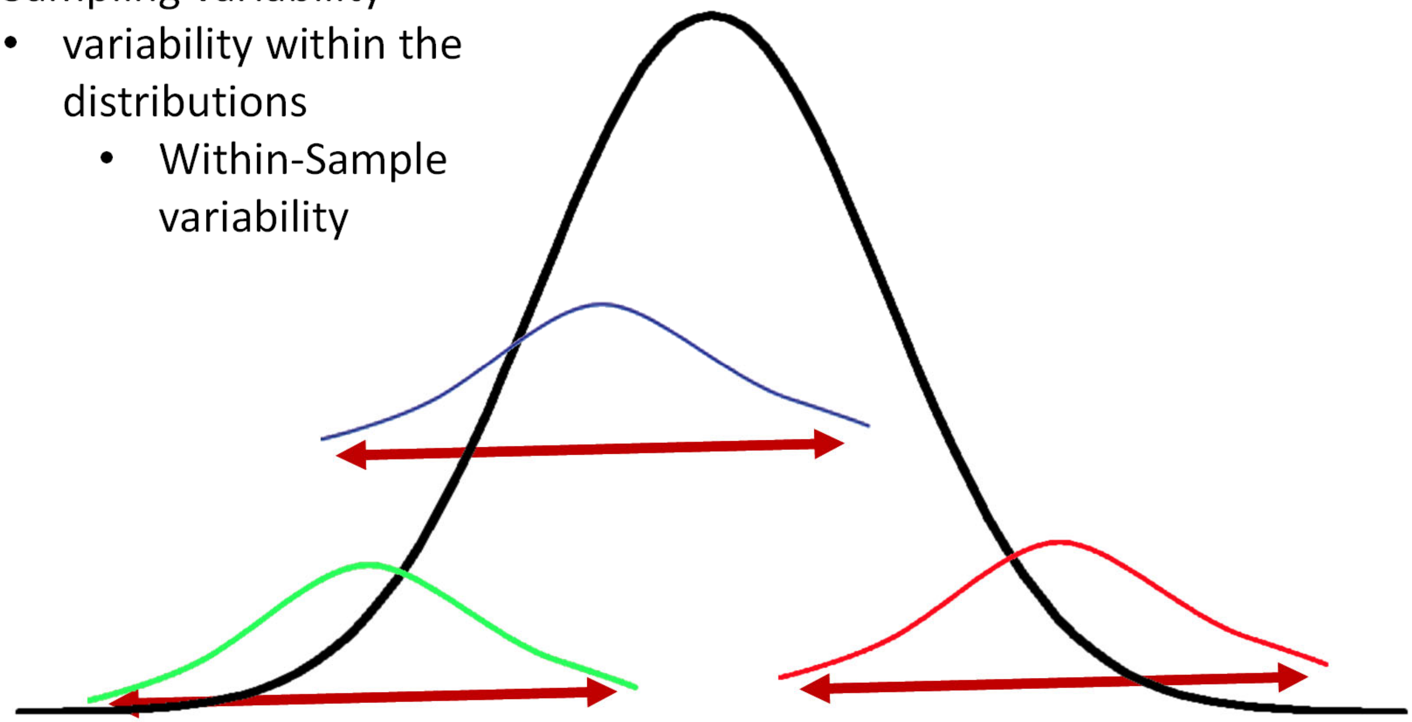
- variability among the sample means
  - Between-Sampling variability



Testing the likelihood that these samples are from the same population

Sampling variability

- variability within the distributions
  - Within-Sample variability

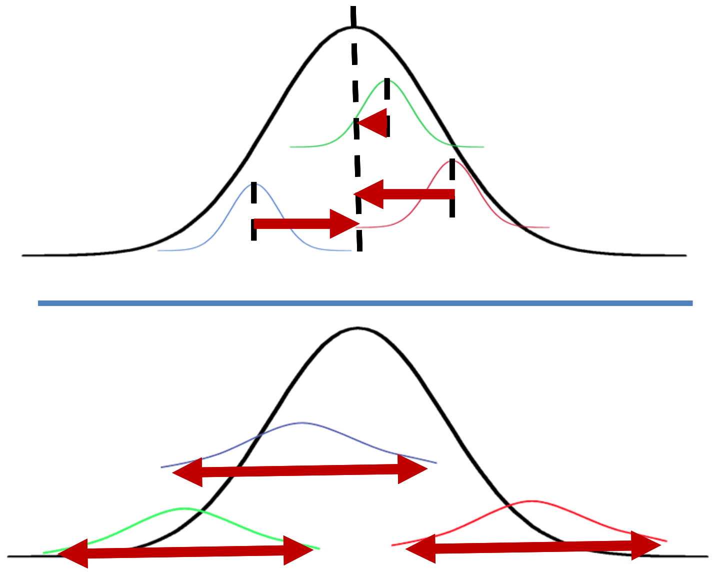


ANOVA =

variability among the sample means

---

variability within the distributions



Signal

Noise

Oneway ANOVA

Data Science: Comparison

- Professor Huang is teaching three data science fundamentals classes. One of these classes is delivered online, the second one is delivered in person on campus, and the third one is a hybrid class. Professor Huang wants to know if there are differences in performance due to the delivery platform.

ANOVA

ANOVA with Post Hoc

```
import pandas as pd
# For oneway ANOVA
import scipy.stats as st

# For Post Hoc
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison as multi

# Read data stored in csv file
df = pd.read_csv("differences3.csv")

# assign label to each student type: Online, InPerson, Hybrid
df["Student"].replace({1:"Online", 2:"InPerson", 3:"Hybrid"}, inplace=True)

# Oneway ANOVA
rdf = st.f_oneway(df["Score"][df["Student"]=="Online"], df["Score"][df["Student"]=="InPerson"], df["Score"][df["Student"]=="Hybrid"])

print ("ANOVA Results: ", rdf)

# Post Hoc
mc = multi(df["Score"], df["Student"])
posthoc = mc.tukeyhsd()

print ()
print (posthoc)
```

Oneway ANOVA

Data Science: Comparison

- Professor Huang is teaching three data science fundamentals classes. One of these classes is delivered online, the second one is delivered in person on campus, and the third one is a hybrid class. Professor Huang wants to know if there are differences in performance due to the delivery platform.

ANOVA

ANOVA

```
import pandas as pd
import scipy.stats as st

# Read data stored in csv file
df = pd.read_csv("differences3.csv")

# assign label to each student type: Online, InPerson, Hybrid
df["Student"].replace({1:"Online", 2:"InPerson", 3:"Hybrid"}, inplace=True)

# Oneway ANOVA
rdf = st.f_oneway(df["Score"][df["Student"]=="Online"], df["Score"][df["Student"]=="InPerson"], df["Score"][df["Student"]=="Hybrid"])

print ("ANOVA Results: ", rdf)
```

ANOVA

```
ANOVA Results:  F_onewayResult(statistic=1.251646103688551, pvalue=0.2937751293444691)
```

If sig (p-value) < 0.05, then we reject null hypothesis. Therefore, we conclude that significant difference exists.

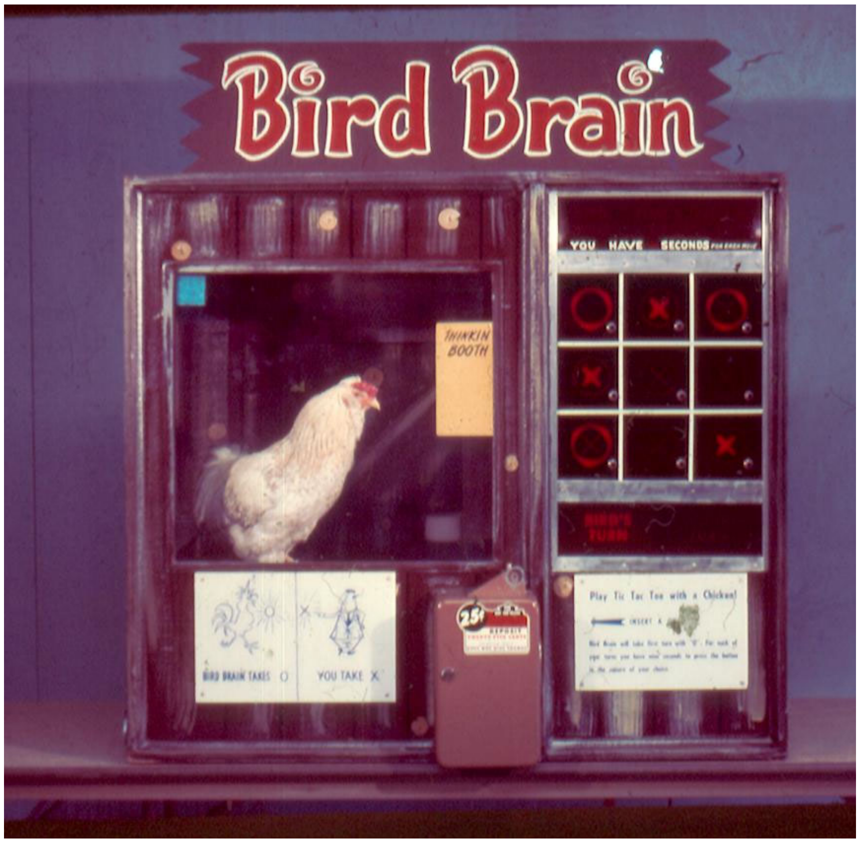
If sig > 0.05, then we accept the null hypothesis.

ANOVA with Post Hoc

```
import pandas as pd
# For oneway ANOVA
import scipy.stats as st

# For Post Hoc
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison as multi

# Read data stored in csv file
df = pd.read_csv("differences3.csv")

# assign label to each student type: Online, InPerson, Hybrid
df["Student"].replace({1:"Online", 2:"InPerson", 3:"Hybrid"}, inplace=True)

# Oneway ANOVA
rdf = st.f_oneway(df["Score"][df["Student"]=="Online"], df["Score"][df["Student"]=="InPerson"], df["Score"][df["Student"]=="Hybrid"])

print ("ANOVA Results: ", rdf)

# Post Hoc
mc = multi(df["Score"], df["Student"])
posthoc = mc.tukeyhsd()

print ()
print (posthoc)
```

Machine Learning

Machine Learning



Machine Learning

Branch of artificial intelligence using data to train a machine (model) to make predictions based on inputs (data)

Machine Learning

$\{2, 4, 6\}$ $\{8\}$

$\{1,8, 22\}$ $\{50\}$

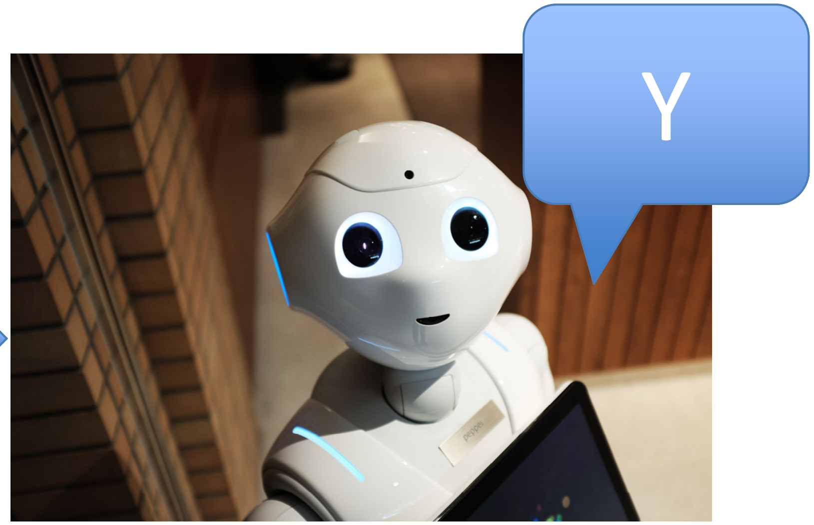
$$\underbrace{\{X_1, X_2, X_3\}}_{\text{Features}} \quad \underbrace{\{Y\}}_{\text{Label}}$$

Machine Learning

{2, 4, 6} {8}

{1, 3, 5} {7}

$\{X_1, X_2, X_3\}$



$$\underbrace{f(X)}_{\text{Feature}} = \underbrace{Y}_{\text{Label}}$$

Machine Learning

- Supervised Learning
  - Data for training machine learning model include known labels (outputs) and features (inputs)
- Unsupervised Learning
  - Data for training model include only features (inputs) but no known labels (outputs)
    - Machine learning model is trained by observing similarities in features (inputs)

Machine Learning

- Supervised Learning
  - Popular supervised learning method
    - Regression model

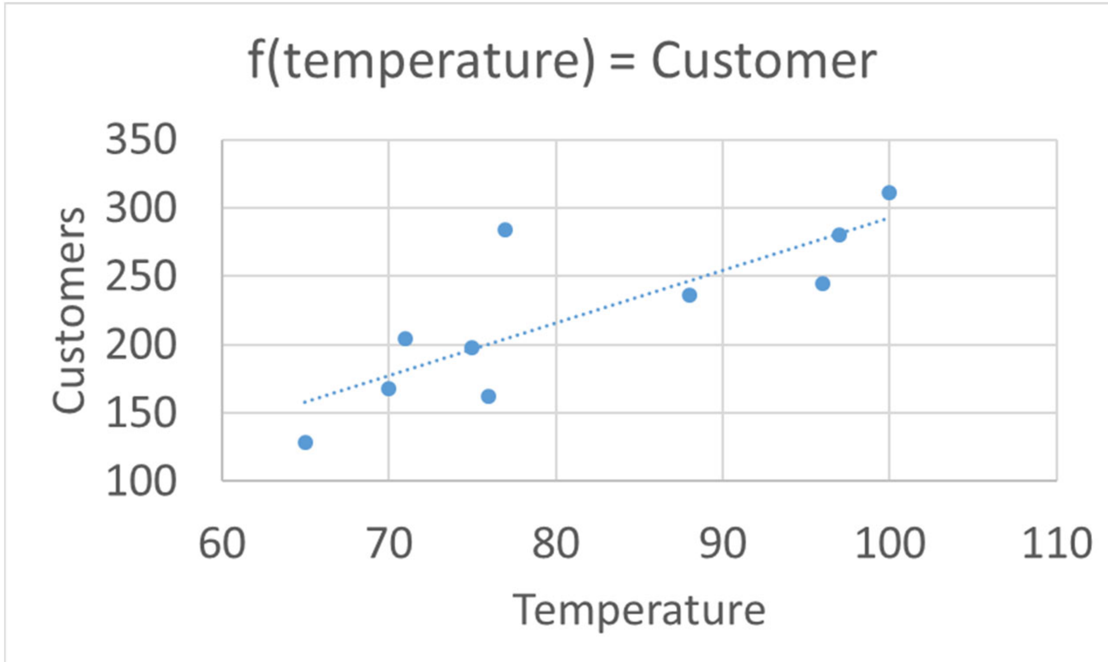
$$f(x) = y$$

$$Y = a + bX$$

where X is the explanatory variable
Y is the dependent variable.
b is the slope of the line
a is the intercept value of y when x = 0)

Machine Learning

Temperature	Customer
71	204
75	198
100	311
65	128
97	280
77	284
70	168
88	236
76	162
96	245



Machine Learning

Machine Learning: Regression Model

Machine Learning: Regression Model

Temperature	Customer
71	204
75	198
100	311
65	128
97	280
77	284
70	168
88	236
76	162
96	245

Machine Learning: Regression Model

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.813790773
R Square	0.662255423
Adjusted R Square	0.620037351
Standard Error	36.81556566
Observations	10

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	21261.313	21261.313	15.68653871	0.004173386
Residual	8	10843.087	1355.385875		
Total	9	32104.4			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	-91.29220104	79.85396655	-1.143239403	0.285994895	-275.4357781	92.85137604	-275.4357781	92.85137604
Temperature	3.839168111	0.969334269	3.960623525	0.004173386	1.603879278	6.074456944	1.603879278	6.074456944

Multiple R

- Absolute value of correlation coefficient (Pearson r)
  - The large the number the more indication of possible relationship
  - Can’t tell the direction because of the absolute value

Machine Learning: Regression Model

R^2

- coefficient of determination
  - How well the regression model (line) fits the data
  - Proportion of the variance in the dependent variable that is explainable (predictable) by he independent variable
  - $R^2 = 1$ means 100% of the dependent variable can be explained by the independent variable
  - $R^2 = 0.80$ means 80% of the dependent variable can be explained by the independent variable

Standard Error

- A measure of the precision of the model
  - Average error of the regression model.
  - Tells how wrong the model is
  - The smaller the better (in relation to the coefficient)

Significant F

- Significant F is the P-value of F
  - a ratio computed by dividing the mean regression sum of squares by the mean error sum of squares
  - Ranges from 0 to very large number
  - Model is OK if less than 0.05
  - Look for another independent variable if greater than 0.05

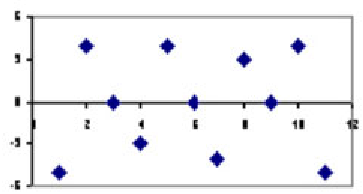
P-values

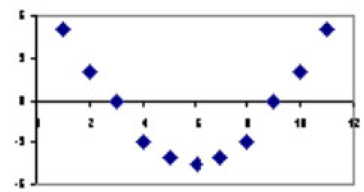
- Probability that the estimated coefficient is unreliable.
  - OK if less than 0.05
  - Otherwise, delete the independent variable > 0.05

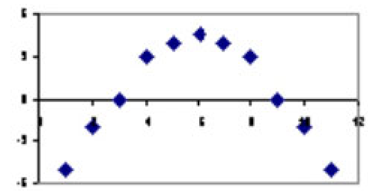
Machine Learning: Regression Model

Residuals

- error = y – ŷ (y actual – y predicted)







Machine Learning: Regression Model

Regression Results

Data Science Fundamentals: Regression Results

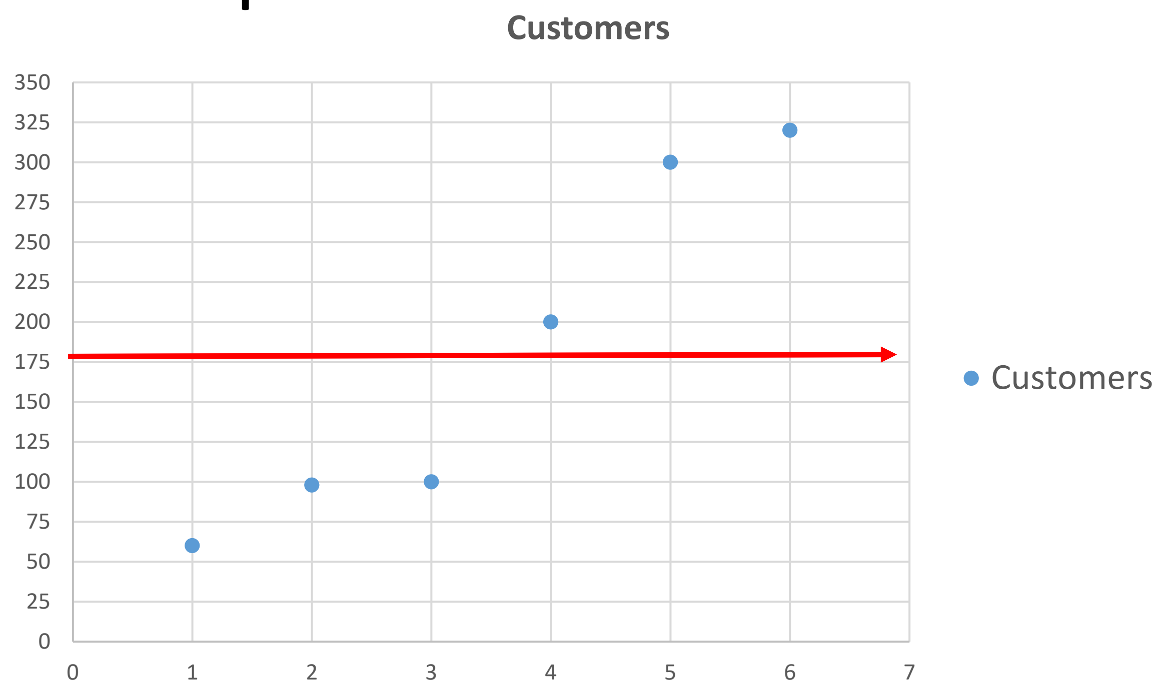
- Temperature vs. Customers

Temperature	Customers
100	60
95	98
90	100
85	200
80	300
75	320

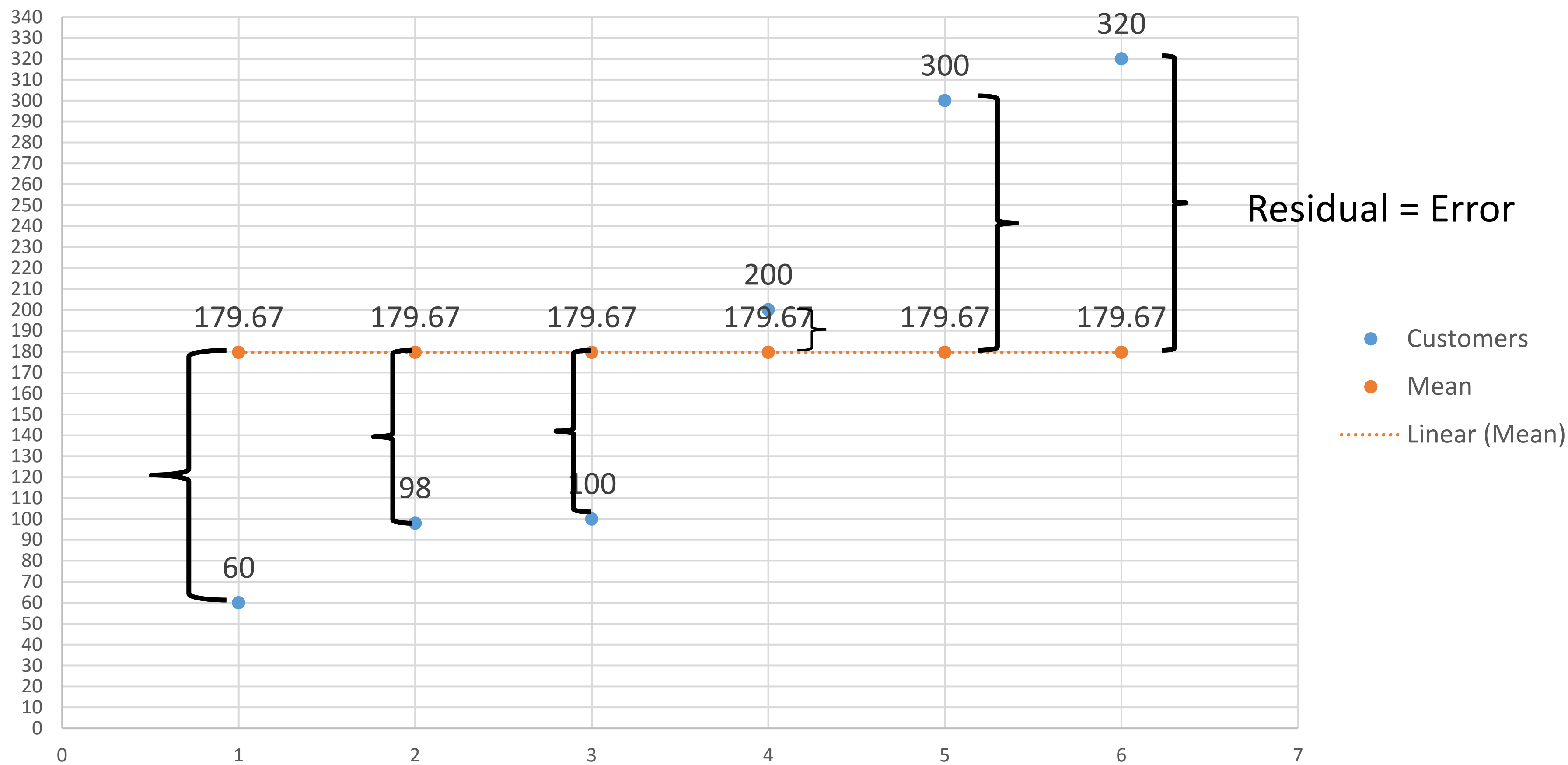
Regression Results

- Customers = Y = Dependent Variable

Y	Ȳ
Customers	**Mean**
60	180
98	180
100	180
200	180
300	180
320	180



Regression Results



Regression Results

ERROR

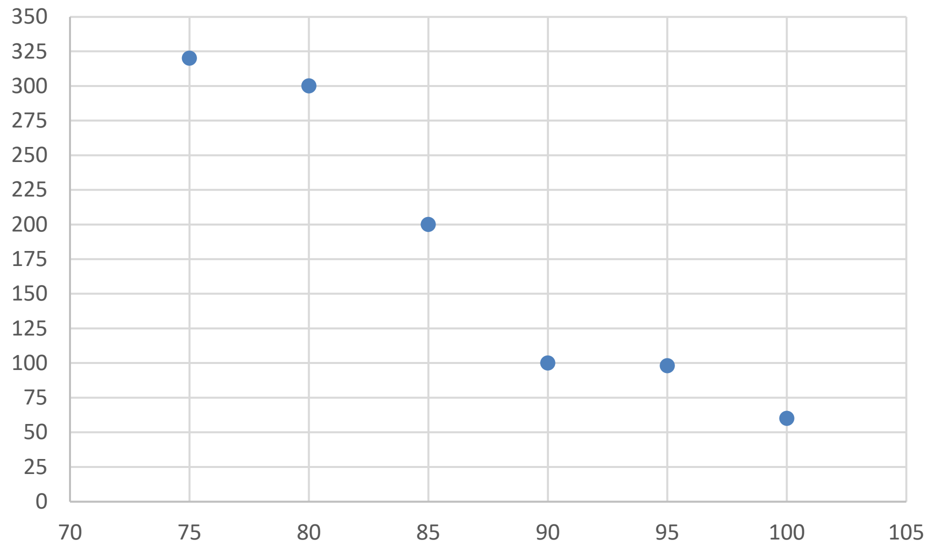
Residual Residual ^{2}

Y	$\bar{Y}$	$Y - \bar{Y}$	$(Y - \bar{Y})$^2
60	179.67	-119.67	14320.11
98	179.67	-81.67	6669.44
100	179.67	-79.67	6346.78
200	179.67	20.33	413.44
300	179.67	120.33	14480.11
320	179.67	140.33	19693.44
			61923.33

Total Sum of Square (SST) = Σ **$(Y - \bar{Y})$^2** = 61923.33

Regression Results

X **Temperature**	Y **Customers**
100	60
95	98
90	100
85	200
80	300
75	320



Regression Results

Regression Statistics	
Multiple R	0.963506714
R Square	0.928345189
Adjusted R Square	0.910431486
Standard Error	33.30579815
Observations	6

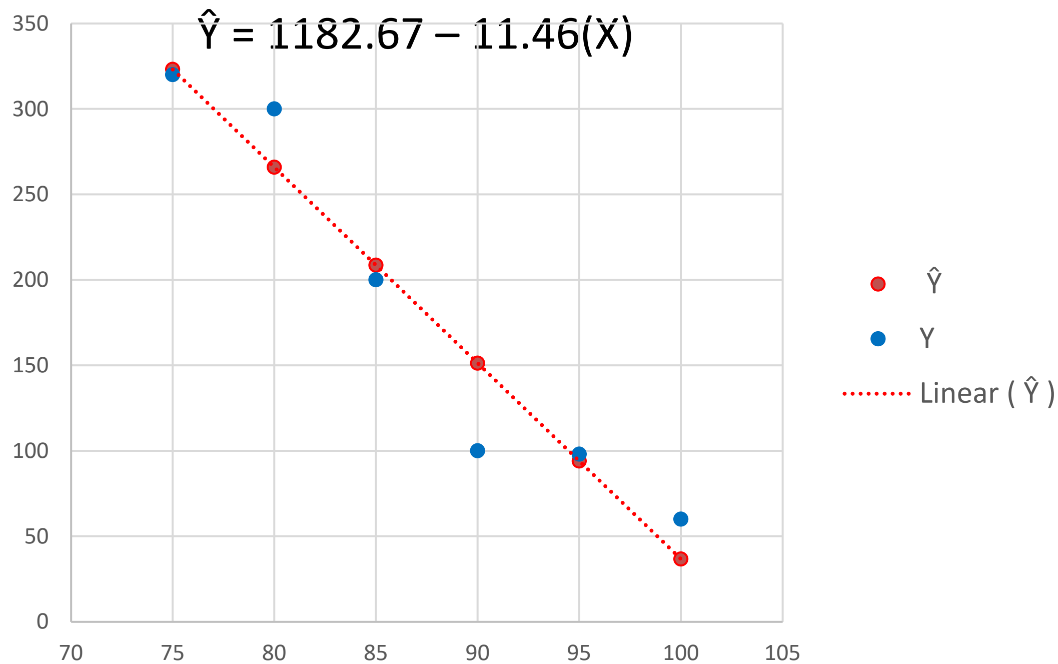
ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	57486.22857	57486.22857	51.82318801	0.00197334
Residual	4	4437.104762	1109.27619		
Total	5	61923.33333			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1182.666667	139.990045	8.448219777	0.001075413	793.9919915	1571.341342	793.9919915	1571.341342
Temerature	-11.46285714	1.592321712	-7.198832406	0.00197334	-15.88385097	-7.041863319	-15.88385097	-7.041863319

$$\hat{Y} = 1182.67 - 11.46(X)$$

Regression Results



X	Y	Ŷ	Y-Ŷ	(Y-Ŷ)^2
100	60	36.67	[illegible]	[illegible]
[illegible]	[illegible]	93.97	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
80	300	265.87	[illegible]	[illegible]
75	320	323.17	[illegible]	[illegible]
Sum of Squares Error (SSE)				4437.49
Total Sum of Squares (SST)				61923.33
Sum of Squares Regression (SSR)				57485.84

SST

SSR

SSE

Regression Results

Regression Statistics	
Multiple R	0.963506714
R Square	0.928345189
Adjusted R Square	0.910431486
Standard Error	33.30579815
Observations	6

R Square = SSR / SST

= 57485.84/ 61923.33

= 0.92834

X	Y	Ŷ	Y-Ŷ	(Y-Ŷ)^2
100	60	36.67	23.33	544.29
95	98	93.97	4.03	16.24
90	100	151.27	-51.27	2628.61
85	200	208.57	-8.57	73.44
80	300	265.87	34.13	1164.86
75	320	323.17	-3.17	10.05
Sum of Squares Error (SSE)				4437.49
Total Sum of Squares (SST)				61923.33
Sum of SquaresRegression (SSR)				57485.84

ANOVA

	df	SS	MS	F	Significance F
Regression	1	57486.22857	57486.22857	51.82318801	0.00197334
Residual	4	4437.104762	1109.27619		
Total	5	61923.33333			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	1182.666667	139.990045	8.448219777	0.001075413	793.9919915	1571.341342	793.9919915	1571.341342
Temerature	-11.46285714	1.592321712	-7.198832406	0.00197334	-15.88385097	-7.041863319	-15.88385097	-7.041863319

$\hat{Y} = 1182.67 – 11.46(X)$

Machine Learning- Python -Regression Analysis

- Temperature vs. Customers

Temperature	Customers
100	60
95	98
90	100
85	200
80	300
75	320

Regression Statistics	
Multiple R	0.963506714
R Square	0.928345189
Adjusted R Square	0.910431486
Standard Error	33.30579815
Observations	6

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	57486.22857	57486.22857	51.82318801	0.00197334
Residual	4	4437.104762	1109.27619		
Total	5	61923.33333			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1182.666667	139.990045	8.448219777	0.001075413	793.9919915	1571.341342	793.9919915	1571.341342
Temerature	-11.46285714	1.592321712	-7.198832406	0.00197334	-15.88385097	-7.041863319	-15.88385097	-7.041863319

$$\hat{Y} = 1182.67 - 11.46(X)$$

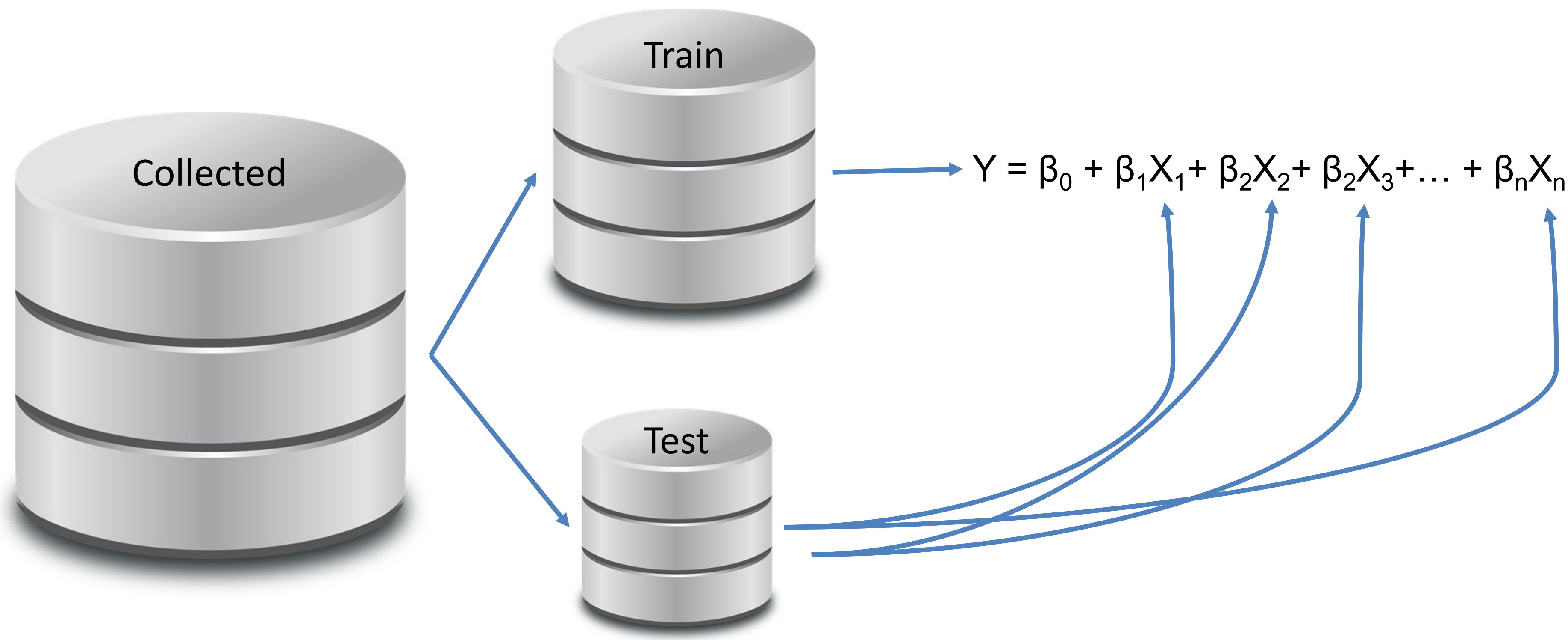
```
import pandas as pd

# need for regression analysis
import statsmodels.api as sm
from statsmodels.formula.api import ols


temperature = [100,95,90,85,80,75]
customer= [60,98,100,200,300,320]

df = pd.DataFrame(temperature, columns=["Temperature"])
df["Customer"] = customer

# Perform Regression Analysis
results = ols ("Customer ~ Temperature", data=df).fit()
print (results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Customer   R-squared:                       0.928
Model:                            OLS   Adj. R-squared:                  0.910
Method:                 Least Squares   F-statistic:                     51.82
Date:                Fri, 29 May 2020   Prob (F-statistic):            0.00197
Time:                        21:50:05   Log-Likelihood:                -28.332
No. Observations:                   6   AIC:                             60.66
Df Residuals:                       4   BIC:                             60.25
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept    1182.6667    139.990      8.448      0.001     793.992    1571.341
Temperature   -11.4629      1.592     -7.199      0.002     -15.884      -7.042
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   1.909
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.477
Skew:                          -0.659   Prob(JB):                        0.788
Kurtosis:                       2.585   Cond. No.                         905.
==============================================================================
```

$$\hat{Y} = 1182.67 - 11.46(X)$$

Linear Regression in Machine Learning: Train and Test Model

Linear Regression in Machine Learning: Train and Test Model



```
import mysql.connector as sq
import pandas as pd

# needed for machine learning regression model training and testing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Connecting to MySQL, query database, store results in dataframe variable
mydb=sq.connect(host="localhost",user="root",passwd="ucla", buffered=True)
query = "SELECT * FROM covid19USA531.covid19USA531"
df = pd.read_sql(query,mydb)
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_tests	new_tests
0	USA	United States	3/14/2020	2174	511	47	7	31732	4575
1	USA	United States	3/15/2020	2951	777	57	10	39332	7600
2	USA	United States	3/16/2020	3774	823	69	12	57173	17841
3	USA	United States	3/17/2020	4661	887	85	16	72856	15683
4	USA	United States	3/18/2020	6427	1766	108	23	97590	24734

```
# prepare x by droping y = total_deaths
x = df.drop(["iso_code", "location", "date", "total_deaths"], axis=1)
```

	total_cases	new_cases	new_deaths	total_tests	new_tests
0	2174	511	7	31732	4575
1	2951	777	10	39332	7600
2	3774	823	12	57173	17841
3	4661	887	16	72856	15683
4	6427	1766	23	97590	24734

```
# prepare y = total_deaths
y = df.total_deaths
```

```
0         47
1         57
2         69
3         85
4        108
       ...
74     98916
75    100442
76    101617
77    102836
78    103781
Name: total_deaths, Length: 79, dtype: int64
```

```python
#train_and_test_data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=40)
```

```python
model = LinearRegression()
model.fit(x_train, y_train)
```

```python
y_predict = model.predict(x_test)
```

```python
model.score(x_test,y_test)
```

```python
model.coef_
```

```python
model.intercept_
```